BABAZ: A LARGE SCALE AUDIO SEARCH SYSTEM FOR VIDEO COPY DETECTION

Hervé Jégou[⋆], *Jonathan Delhumeau*[⋆], *Jiangbo Yuan*[•], *Guillaume Gravier*[†], *Patrick Gros*[⋆]

[⋆]INRIA [•]Florida State University [†]CNRS

ABSTRACT

This paper presents BABAZ, an audio search system to search modified segments in large databases of music or video tracks. It is based on an efficient audio feature matching system which exploits the reciprocal nearest neighbors to produce a per-match similarity score. Temporal consistency is taken into account based on the audio matches, and boundary estimation allows the precise localization of the matching segments. The method is mainly intended for video retrieval based on their audio track, as typically evaluated in the copy detection task of TRECVID evaluation campaigns. The evaluation conducted on music retrieval shows that our system is comparable to a reference audio fingerprinting system for music retrieval, and significantly outperforms it on audio-based video retrieval, as shown by our experiments conducted on the dataset used in the copy detection task of TRECVID'2010 campaign.

***Index Terms*—** audio fingerprinting, audio search, copy detection, reciprocal neighbors, TRECVID

1. INTRODUCTION

Due to the emergence of networks and personal capture devices, large volumes of image and video are now available in digital format, which has enabled new services and easier access to large amount of content. On the downside, the marginal cost of copying a video is extremely low. In conjunction with the generalization of high-speed Internet connection, a pirate can easily and massively diffuse illegal content on Internet, in particular through peer-to-peer networks. This is a threat which has a critical impact on copyright holders and artists, which have less revenues because the content is used without being paid for. Moreover, the videos may be uploaded on websites hosting user-generated content (UGC), such as Youtube or Dailymotion. These companies are likely to host illegal content without knowing it, given that the mount of videos uploaded every day forbids a manual verification.

For these reasons, automatic detection of pirated videos on peer-to-peer networks and UGC websites has received a significant attention from the multimedia and signal processing communities in the last few years, in particular using the image channel [1]. Research on that topic is supported by the National Institute of Standard and Technology (NIST), which since 2008 organizes a specific task in TRECVID [2] to evaluate the different techniques. However, as observed in TRECVID's 2010 evaluation campaign, most of the systems only consider the visual channel, or use off-the-shelf solutions adapted from the audio community [3], such as the one proposed by Wang [4] in the original Shazaam audio fingerprinting system or the Echoprint project[1]. Although the current Shazaam implementation probably departs from that of the original paper, a Matlab implementation by Dan Ellis of this method is available online [5] and will be used as the reference method for this reason.

In this paper, we propose an audio fingerprint system, called BABAZ, that is intended for video copy detection based on audio, unlike most concurrent schemes which were mainly addressing the problem of query-by-example on mobile devices [5, 6]. It has been successfully used for the copy detection task of the TRECVID'2011 campaign, see [7]. It is available online[2] under open-source license.

BABAZ is inspired by state-of-the-art image search systems. The first step extracts descriptors from the waveform, which are treated as sequences of Euclidean descriptors. This departs from [4], based on landmark hashes. The nearest neighbors of the query descriptors are retrieved using an indexing structured shared by all audio tracks, and compute a similarity measure that takes into account the approximate *reciprocal neighborhood* of the selected nearest neighbors. The temporal consistency is exploited through a temporal Hough transform, similar to [4] and visual video search systems [1, 8]. Finally, we introduce the idea of time-shifting on query side. It consists in submitting to the system several versions of the query, shifted by a few milliseconds. This approach, which approximately multiplies the query time by the number of versions considered, increases the recall by avoiding the cases where the query and database videos are sampled in phase opposition.

This paper is organized as follows. Section 2 describes the main components of BABAZ and Section 3 reports our experiments in two application scenarios: the dataset used in TRECVID'2010 copy detection task and a music landmark detection setup as proposed in [4]. Our experiments demonstrate the better performance of our system for video copy detection.

2. BABAZ AUDIO SEARCH SYSTEM

This section describes the main components of BABAZ. We will consider two application cases, leading to two variants of our system, namely BABAZ_v and BABAZ_m:

- BABAZ_v considers the application case for which our system was originally designed, i.e., a copy detection setup such as the one considered in TRECVID [2], where the signal is the audio track of a video. It typically includes voices, silences and occasionally music. The audio may have been transformed by different kinds of transformations, such as strong pass-band filter, compression, mixing, single- or multi-band companding, etc.
- BABAZ_m considers an application scenario comparable to that of [4], i.e., music retrieval in a noisy environment. This setup is considered, in particular, for the sake of comparison with [4] in their own application scenario.

BABAZ_v and BABAZ_m only differ in the choice of the filter banks, as detailed in the feature extraction step described below.

[1]http://echoprint.me

[2]http://babaz.gforge.inria.fr

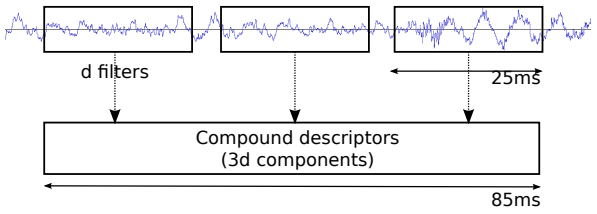

Fig. 1. Compound descriptor.

2.1. Pre-processing

The audio tracks extracted from an audio corpus are not necessarily homogeneous. Sample rates as well as encoding quality vary significantly from one track to another. It is in particular the case in the Internet Archive dataset used in the TRECVID's copy detection task, where the videos are mainly amateur videos captured and encoded by different devices and audio codecs. In order to deal with this variability in a consistent way, all the tracks are resampled to 32,000 Hz. We use the right stereo channel only when stereo is available.

2.2. Feature extraction: filter banks

Hereafter, we detail how we extract descriptors from the audio waveform. The signal is ran through a pre-emphasis filter to compensate for the spectral slope and divided into overlapping short-term windows of 25 ms taken every 10 ms. In each window, the short-term spectrum is represented by log-energies at the output of overlapping band-pass filters. As mentioned above, two variants are considered:

- BABAZ$_{\text{v}}$: For video copy detection, the signal is severely attacked, and may have undergone a strong band-pass filter. In this case, we use 40 filters spread along the [500 Hz,3000 Hz] frequency range on a Mel scale.
- BABAZ$_{\text{m}}$: In music retrieval, the audio tracks have a relatively good frequency localization, which is exploited in [4]. In this application scenario, we consider 64 filters spread along the [200 Hz,4000 Hz] frequency range (Mel scale).

As a result, the dimensionality d of the descriptors is either $d = 40$ or $d = 64$. The representation based on these filters gives a rough approximation of the signal's spectral shape in the frequency range considered while smoothing out the harmonic structure, if any, and is therefore robust to many spectral distortions. We have used the freely available `spro` software[3] for the generation of filter banks. This software also includes an efficient implementation of the widely used MFCC descriptors. However, in our experiments, these descriptors are significantly outperformed by the filter banks.

2.3. Compound descriptors and energy invariance

The temporal consistency provided by a single filter bank is limited, as their temporal span is limited and only frequencies are considered. This is problematic since the database is large: the filter banks themselves might not be discriminative enough to identify a matching hypothesis with sufficient reliability.

In order to increase the discriminative power of the descriptor, the temporal aspect is emphasized by concatenating several filter banks, as done in Serra's thesis [9] in a context of cover detection.

For a given timestamp t, 3 successive filter banks are extracted at timestamps $t - \delta$, t and $t + \delta$, producing a compound descriptor of dimensionality $3d$ (i.e., 120 or 192, depending on the application scenario). We set $\delta = 30$ ms in order to avoid overlapping. We have performed a few experiments on a validation dataset to decide on how to take into account this dynamic aspect, e.g., using derivatives of the filter bank with respect to time. Compounding the descriptors appeared a reasonable choice. As illustrated in Figure 1, the resulting span of this descriptors is 85 ms. This approach favors the temporal aspect by taking into account the dynamic behavior of the frequency energies, at the cost of an increased descriptor dimensionality.

Descriptors are compared with the Euclidean distance. For large vector databases it allows for efficient indexing algorithms, as the one described in the next subsection. Note however that we have not performed a thorough evaluation of other possible distances/divergences because the search algorithms are generally less efficient for sophisticated metrics. In particular, we have not considered the metrics specifically developed to compare filter banks, such as the Itakura-Saito divergence or the log-spectral distance,

In order to take into account attacks on the volume (signal energy), the descriptor is finally made invariant by subtracting its mean.

2.4. Approximate nearest neighbor search

As the exact search is not efficient enough, BABAZ uses an approximate nearest neighbor search technique. Many methods exist for this task, such as locality sensitive hashing or the FLANN package [10]. However, this step has a major impact on both efficiency and search quality, and only a few methods are able to search in hundreds of millions of descriptors with reasonable quality, as required by our method to index thousands of hours of audio.

BABAZ uses the IVFADC indexing method of [11], which is able to index billions of descriptors on a commodity server. It finds the approximate k nearest neighbors using a compression-based approach, and relies on an inverted structure to avoid exhaustive search. This approximate nearest neighbor method implicitly sees multi-dimensional indexing as a vector approximation problem. It is proved [11] that the square error between the distance and its estimation is bounded, on average, by the quantization error. This ensures, asymptotically, near perfect search results when increasing the number of bits allocated for the quantization indexes:

The main parameters of this method are the number of bytes b used per database vector and the number c of inverted lists associated with the partitioning of the feature space (learned by k-means). In our case, we set $b = 24$ bytes and use multiple assignment [11] on query side, leading to visit 16 inverted lists out of $c = 16,384$.

2.5. Scoring vote and reciprocal nearest neighbors

The search technique returns a list of k (approximate) nearest neighbors. A conventional method to exploit them consists in assigning a vote of 1 to all the corresponding audio tracks, or alternatively, a function of the rank or of the distance. Based on a recent state-of-the-art work [12] in image search, we adopt a different strategy, which is illustrated in Figure 2.

Denoting by $d_k(q)$ the distance between the query descriptor and its k-th nearest neighbor, the quantity $d_k(q) - d(q, i)$ is shown, based on a mutual information criterion [12] measured on image descriptors, to better reflect the quality of the match. This is also the case for our audio descriptors, so we adopt this weighting scheme.

The distance $d_k(q)$ is relative to the query. In order to symmetrize the relationship between the query and database descriptors, it is worth considering the *reciprocal* nearest neighbors of the database vector, or more specifically the typical distance between the database vector and its own k-nearest neighbors.

[3] http://gforge.inria.fr/projects/spro

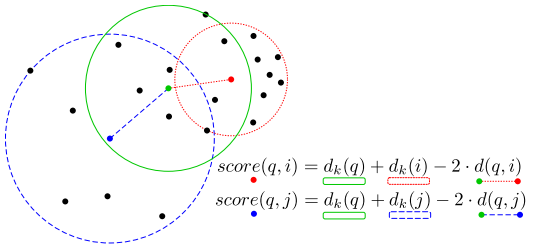

Fig. 2. Reciprocal nearest neighbors and of our voting strategy.

In practice, computing the reciprocal nearest neighbors is impractical: the audio descriptor database may contains up to billions of vectors. If exact nearest neighbor search is used, then it turns out that each database vector has to be submitted to the system. Although some approximate strategies [13] were proposed to compute the nearest neighbor graph, these approaches were only tested on up to 1 million vectors. However, we are not interested in the neighbors themselves, but in the typical distance of a database vector to its neighborhood. This reciprocal typical distance is estimated on a limited subset of 1 million vectors. In this case, the parameter k associated with the database vectors has to be adjusted to account for the smaller size of this subset.

Re-ranking: Finally, in the spirit of [14], the hypotheses are re-ranked based on exact descriptors to obtain the exact distances, in order to increase the precision of the proposed similarity. The difference with [14] is that we use the original descriptors and not only a compressed version of these.

Energy weighting: Video tracks contain many silences. Those are filtered when the signal and consequently the descriptor is zero. However, there are also many descriptors extracted on audio frames containing almost no energy, but which are not pure silence. Filtering audio segments with low energy may lead to loose some precious information, and reduce the accuracy of the localization. For this reason, we adopt a smoother strategy and multiply the score associated with the match with the energy of the query descriptor.

2.6. Hough matching

BABAZ assumes that the transformations do not include any acceleration. Given a query, for each of its audio descriptors we first search for the k approximate nearest neighbors and compute their weighting score based on the strategy exposed above. We then vote for several alignment hypotheses (a_b, Δ_T) using the scoring method introduced above. Compared with uniform voting, this brings a slight improvement at almost no cost in efficiency. The video hypotheses with low scores are filtered. On output, this Hough matching system returns a maximum of 40 hypotheses per query. Each database track is associated with a maximum of 3 Δ_T hypotheses.

2.7. Detection of boundaries

At this point, for each query we may have several alignment hypotheses (id, Δ_T), where id is the database track identifier and Δ_T is the difference between the query and the database timestamps. We use the whole set of descriptors and weight their distance to produce a score per time instant. This score is filtered in time and used to detect the boundaries defining a matching segment. Each segment receives a score computed from the individual scores for each time instant.

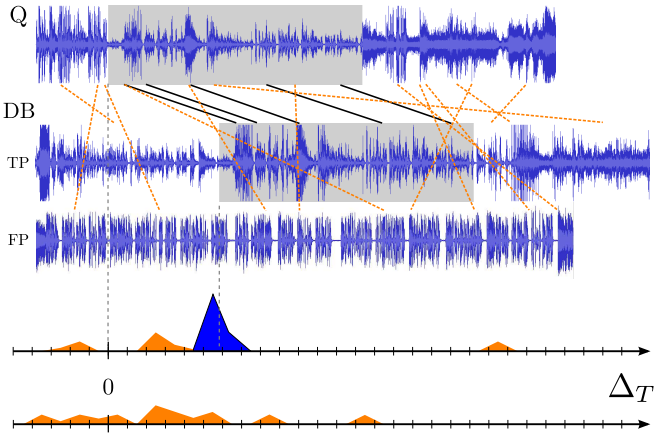

Fig. 3. Illustration of the temporal Hough transform: the audio matches output by the approximate search engine are collected and summed up for each hypothesis (id, Δ_T). This dilutes the scores of false positives over time shift hypotheses.

2.8. Shifted query

The audio descriptors are extracted every 10 ms, which leads to reduce the quality of the comparison if the sampling of the database track occurs in phase opposition, i.e., with a shift of 5 ms relative to the query track. To address this problem, we submit several shifted version of the query to the system. For instance, we create shifted versions of the query with shifts of 2, 4, 6 and 8 ms. This, obviously, significantly impacts the efficiency of the search by a significant factor, and should be used when high precision is required only.

3. EXPERIMENTS

We have conducted some experiments in two applications cases: audio landmark detection, and copy detection. Our system is compared with the approach of [4], using the Matlab implementation by Dan Ellis [5] available online. In order for this approach to scale to the larger datasets we consider here, we have adapted the size of the hash table to avoid collusions.

3.1. Video copy detection based on audio

For audio-based video copy detection, we have considered the dataset provided for the TRECVID'2010 copy detection task, as TRECVID can now be considered as the standard evaluation setup for this application scenario. It comprises more than 11,000 video clips corresponding to about 400 hours of video. The queries are transformed excerpts (3 to 30 seconds) of some database video, and are inserted in a longer distractor track. About 1/3 of the queries have no corresponding video in the database. In this case the system should return no result. Our optimization has been performed on an independent validation dataset.

The comparison of BABAZ$_v$ with [4] is shown in Figure 4 using a precision-recall curve. Our system is significantly better in this application scenario. Our system is 13 times faster than real time on a 24-core machines without the shift extension. This is slower than the Wang's system. The shift provides a relatively low improvement given the significantly increased complexity. It is however worth using it in an evaluation like TRECVID, where the recognition performance is the most important measure.

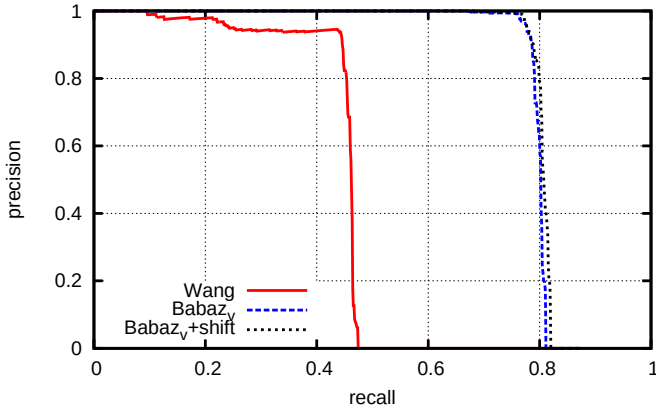

Fig. 4. Precision recall on Trecvid'10.

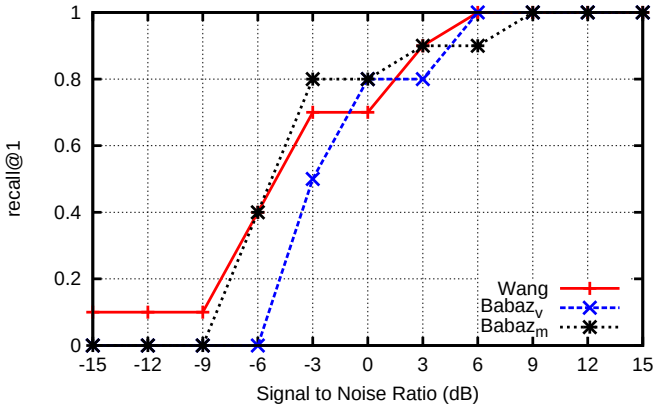

Fig. 5. Recognition rate (correct track ranked in first position) on the music retrieval dataset with varying amount of noise. Query audio segments are 10s long.

3.2. Music detection

In this evaluation, we have considered an audio dataset comprising 1,000 tracks (approximately 55 hours) of mixed sound files (music, voice and other) extracted from TRECVID'dataset. We have considered an evaluation protocol similar to that of [4]: 10 musical queries of various genres cropped to individual length of 5, 10 or 15 seconds. The energy of the signal is normalized before being mixed with a distractor consisting of crowd noise from a pub (also normalized). The respective signal to noise ratios are between -15dB an 15dB for the set of 330 queries. The performance of our system BABAZ$_v$, and its variant BABAZ$_m$ dedicated to music retrieval, are first shown in Figure 5 for queries of 10s. Note that our result for the Wang's system [4] are consistent with those reported in his paper. Our system reports comparable performance.

Another set of 330 queries were extracted by applying GSM 6.10 compression and 8kHz subsampling to the first query set. The results are presented in Figure 6 with respect to precision-recall, our system is much better. The discrepancy with Figure 5 is explained by the fact that the scores, in our system, are better normalized across queries, which is taken into account by precision-recall (which interleaves the hypotheses of all queries based on the recognition score) but not by the recognition rate.

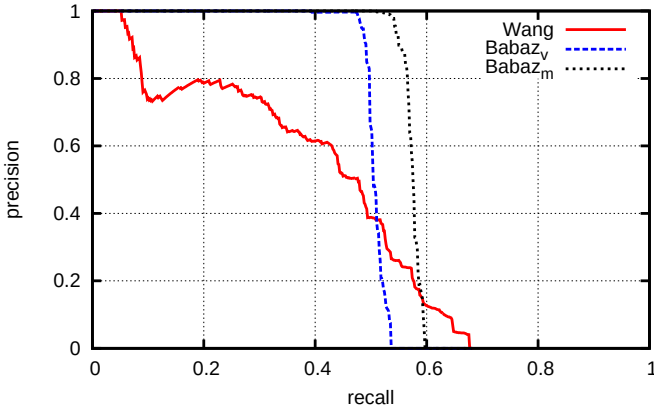

Fig. 6. Precision recall on the music retrieval dataset.

4. CONCLUSION

We have presented the BABAZ audio search system, an open system dedicated to video copy detection based on audio track. It also exhibits competitive search quality for landmark retrieval, although in this context it is slower than the very efficient approaches based on spectral peaks.

5. REFERENCES

[1] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford, "Video copy detection: a comparative study," in *CIVR*, New York, NY, USA, 2007, pp. 371–378, ACM.

[2] A. F. Smeaton, P. Over, and W. Kraaij, "Evaluation campaigns and trecvid," in *MIR*, 2006, pp. 321–330.

[3] P. Cano, E. Battle, T. Kalker, and J. Haitsma, "A Review of Audio Fingerprinting," *The Journal of VLSI Signal Processing*, vol. 41, no. 3, pp. 271–284, 2005.

[4] A. L.-C. Wang, "An industrial-strength audio search algorithm," in *ISMIR'03*, 2003.

[5] D. Ellis, "Robust Landmark-Based Audio Fingerprinting," 2009, `http://labrosa.ee.columbia.edu/matlab/fingerprint`.

[6] V. Chandrasekhar, M. Sharifi, and D. A. Ross, "Survey and evaluation of audio fingerprinting schemes for mobile query-by-example applications," in *ISMIR*, 2011.

[7] M. Ayari, J. Delhumeau, M. Douze, H. Jégou, D. Potapov, J. Revaud, C. Schmid, and J. Yuan, "Inria@trecvid'2011: Copy detection & multimedia event detection," in *TRECVID Workshop*, December 2011.

[8] M. Douze, H. Jégou, and C. Schmid, "An image-based approach to video copy detection with spatio-temporal post-filtering," *IEEE Trans. on Multimedia*, vol. 12, no. 4, pp. 257–266, jun 2010.

[9] J. Serrà, *Identification of versions of the same musical composition by processing audio descriptions*, Ph.D. thesis, Universitat Pompeu Fabra, Barcelona, 2011.

[10] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *VISAPP*, February 2009.

[11] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 33, no. 1, pp. 117–128, January 2011.

[12] H. Jégou, M. Douze, and C. Schmid, "Exploiting descriptor distances for precise image search," Research report, INRIA, June 2011.

[13] W. Dong, M. Charikar, and K. Li, "Efficient k-nearest neighbor graph construction for generic similarity measures," in *WWW*, 2011.

[14] H. Jégou, R. Tavenard, M. Douze, and L. Amsaleg, "Searching in one billion vectors: re-rank with source coding," in *ICASSP*, Prague Czech Republic, 2011.